I'm not a bot

React js advanced tutorial pdf

React js tutorial beginner to advanced. React js tutorial advanced. React advanced tutorial. React basic to advanced. React js beginner to advanced.

Level up your React and Next.js skills with advanced concepts about SSR, PWA, and more! This book covers: * Latest and core React concepts, including hooks and reconciler * Server-Side Rendering (SSR) with React and how to use it with Next.js * Progressive Web Apps (PWA) in React and how to create them * Intermediate and advanced React concepts, like state management * An overview of React for beginners, covering core concepts and bleeding-edge features Key takeaways include: * Understanding what React is and how to get started * Modern ways to code React applications * Implementing Server-Side rendering with/without Next.js on top of the React library * Working with Progressive Web Apps in React * How React works under the hood * The future of React and bleeding-edge React tech you can use today This book is for readers who have a decent understanding of JavaScript, HTML, CSS, and possibly some experience with React. CSE at BITS Pilani, Mehul Mohan is a renowned expert. As a WWDC'19 Scholar, he founded codedamn, an online learning platform focused on coding. With over 100,000 subscribers on his YouTube channel, codedamn, he creates programming tutorials. His contributions as a security researcher have earned him recognition from companies like Google, Microsoft, and Sony. To cancel your subscription, simply go to the account section in settings and click on "Cancel Subscription." Your membership will remain active until you've paid for the remaining time. Learn more [here](. Perlego offers mobile-responsive ePub books that are available for download via the app. Most PDFs can be downloaded as well, with the remaining ones expected to become downloadable soon. Both plans grant full access to the library and Perlego's features. The main difference is the price and subscription period: the annual plan saves around 30% compared to the monthly plan for 12 months. ###TEXT-TO-SPEECH*** Perlego supports text-to-speech, allowing you to listen to books with the read-aloud symbol. You can pause, speed up, or slow down the reading. ###ADVANCED WEB DEVELOPMENT*** Yes, Advanced Web Development with React by Mehul Mohan is available in PDF and ePUB formats. Perlego offers over one million books for exploration. ###JAVASCRIPT*** JavaScript is the programming language of the web, enabling developers to create increasingly better user experiences through interactivity, animations, effects, and stability. As it progressed, frameworks like React emerged to facilitate faster development and better stability. React is often misunderstood as a mere user interface library, but it's so much more than that. It has the potential to power complex web applications like Facebook, making it an ecosystem in its own right. This tutorial will cover both the basics and intermediate to advanced topics, giving you a comprehensive understanding of React and enabling you to build production-ready applications. At its core, React is a UI library that lets developers create and maintain what users see on their screens. It's designed to be flexible, allowing you to "cherry-pick" the features you need for your project. By breaking down complex interfaces into smaller components, React makes it easier to write clean, maintainable code. In this introductory chapter, we'll explore how React works as a UI library and the declarative approach that sets it apart from other programming paradigms. We'll also touch on imperative programming and why it's essential to understand both approaches. With React's component-based architecture, you can write more efficient code that's easier to maintain. Let's start with a simple example. When coding, you have two options: specify what you want to do (declarative) or how you want to do it (imperative). In this case, we'll focus on declarative programming and see how React makes it easier to build better UI systems. Declarative programming has its own set of advantages and disadvantages. The main benefit lies in its ability to utilize highly optimized algorithms. Since the "how" is abstracted from the end-user, declarative code can tap into complex code that provides significant performance boosts while still offering an easy-to-use interface for users. This approach also promotes code scalability, reduces bugs, and makes the code more understandable. React is built on this declarative model. Instead of specifying how to achieve a particular outcome, you simply tell React what needs to be done. Under the hood, React manages everything in an optimized manner. As we delve deeper into the React ecosystem, it's essential to grasp some fundamental JavaScript concepts that can help with React development. One such concept is understanding 'this' behavior within functions and arrow functions. The 'this' keyword refers to how a function is called, except for arrow functions where its scope is lexical. This distinction can be seen in the example provided, where 'myNameButArrow' is an arrow function that logs 'person1.myname'. Another crucial aspect of JavaScript is the use of '.map', which allows you to manipulate arrays without mutating them. It works by returning a new array with the desired transformations applied to each element. Moreover, when dealing with objects and functions, it's essential to understand how they are called or invoked. For instance, calling 'myFunction' on an object 'obj' will return the original object, while calling 'coolFunction', which is actually referencing the global window object, returns the entire window object. This subtlety in function invocation can impact your code's behavior and outcome. As we proceed with React development, keeping these JavaScript nuances in mind will be crucial for writing efficient, readable, and bug-free code. The article discusses how arrays returned from a function passed in `map` are composed of elements. It provides an example of using `.map` to transform the array: ```javascript const component = arr.map(name => arr.name); console.log(component); // logs ["Mehul Mohan", "James Paul"] console.log(arr); // logs original object ``` This code demonstrates that `map` does not modify the original array. On the other hand, `.reduce` allows reducing an array to a single value using an accumulator and current value. The article highlights the importance of mastering React component patterns for creating reusable components with flexibility and power. It touches on two talks: "Bleeding Edge React Advanced React Component Patterns workshop @ PayPal" and "Mixed Mode React".